STRUCTURED DATABASE SYSTEM TOGETHER WITH STRUCTURE DEFINITION FRAME STORING DOCUMENT BODY DATA

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to structured database systems, and more particularly to a structured database system that uses structured information in documents to manage the documents. In general, information in documents is innately structured. For example, documents having such structured information are documents used in the activities of companies, such as drawings and specifications.

2. Description of the Prior Art

Generally, information in documents has a tree structure as shown in FIGS. 1A and 1B. FIG. 1B schematically shows the tree structure shown in FIG. 1A. Information in documents is structured in terms of structural units or elements such as a group of documents, documents, chapters, sections and paragraphs. The tree structure of a document may be dynamically changed. For example, the tree structure may be expanded by adding a new unit after an existing unit of tree structure or grouping a number of existing units. For example, items are defined in paragraphs, drawings and tables, and are then collected to form a group that follows an existing section.

The structured database handles electronic information of structured documents. The electronic information of documents can be in the form of text data, graphic data (image data and vector data), source code (normally character data), the internal code (normally vector data) of a CAD (Computer Assisted Design) system and so on.

Conventionally, a word processor, a DTP (Desk Top Publishing) system, a CAP (Computer Assisted Publishing) system, and a CAD system are known as devices for creating and managing the electronic data of documents. Further, existing database systems such as an RDB (Relational DataBase), can be used to store and manage documents.

The devices as mentioned above are classified into two types; a first type in which a document is handled as groups of symbols such as characters, control symbols, graphic symbols, or a second type in which a mark called "tag" is added to elements in a document. The devices of the first type handle a document as simple data and therefore have a difficulty in management and reuse of the information structure. For example, it is necessary to perform information retrieval in order to know specific information in a specific document or the history of modified portions. Generally, it is very difficult to correctly obtain all of necessary information by means of the information retrieval for the above-mentioned purpose. Even in a case where the document management table electronically cooperates with documents, it is only possible to retrieve a storage area in which the target document is stored, and it is impossible to correctly obtain necessary information from the target document unless the operator actually sees the contents of the documents.

The devices of the second type are capable of performing management based on the structures of documents. However, the devices of the second type still handle files with documents as groups of blocks of data independent of the structures of the documents, and hence need a particular mechanism like the document management table in order to perform development, management and reuse of documents

(including groups of documents mutually associated) and to perform information retrieval. The above particular mechanism is not directly related to information bodies themselves as in the case of use of papers. Hence, the devices of the second type do not have sufficient efficiency and reliability in information retrieval and so on.

The existing database systems have structures that are optimized for specific operations and do not have the functions of efficiently and effectively supporting the document structures. Hence, the existing database systems have the following disadvantages, particularly, regarding the way that database systems are used.

When a document is stored and managed in an existing database system, the document may be arranged on the basis of the structure thereof. For example, when a document is stored in the RDB system, the document is required to be arranged and stored in the form of a table.

On the other hand, if an existing database system is modified in order to match the structure of a document to be stored and managed, some definitions which were not originally prepared may be defined in the existing database system. For example, it is required to define a pointer for accessing a file and/or a free field for each field of the RDB system. Such an additional definition in the existing database system may degrade the original performance thereof, particularly regarding the efficiency in information retrieval and storage capacity. In some cases, the additional definition may prevent use of the original accessing method, such as a standard query language for the RDB system. Such a problem further degrades the efficiency in accessing the database and sometimes requires a particular remedy, i.e., program, for access.

The structure of documents is flexible. For example, the structural units or elements of documents, such as the numbers of chapters and sections are variable, and the document structure expanded. Normally, the structure definition (schema definition) of the existing database systems is determined before data is actually stored. Hence, it is very troublesome to modify the structure of the active database system when in use. When the active database system is modified, a data backup process will be needed, and the saved data may be required to be loaded into the system again after the modification is complete in order to match the saved data with the modified database structure.

It is required that the database system always stores the latest information regarding documents. When a document is revised, a revised version or edition of the document is issued. In some cases, it will be required to save not only the revised version but also the previous versions made in the past. Hence, it is necessary to efficiently manage documents having a number of versions.

The conventional database systems are easily capable of managing the latest version but need to save the previous versions independent of the latest version. In this case, a particular mechanism such as a register system is needed to manage the correspondence among the latest version and the previous versions. Hence, it is necessary to save all the versions and manage and update the correspondence among the versions.

However, it is practically impossible to manage the correspondence among the versions by means of the register mechanism. For example, if there is a need to reflect an error found in a version to the other versions, it will be very difficult to efficiently access such an error in each of the other versions. Further, there is a possibility that the above error may not be completely corrected in some other versions.